```
//[追蹤試算表中的儲存格/Spreadsheet Tracking]
#define IN "P0407IN.txt"
#define OUT "P0407OUT.txt"
//**************************************
#include <iostream>
#include <time.h>
using namespace std;
void redir(void);
//**************************************
/* Work Space*/
struct command{
    char c[3]; //"EX", "DC", "DR", "IC", "IR"
    int r1, c1, r2, c2;
    int a;
    int x[10];
}cmd[1000];
int r, c, n;
int simulate(int *r0, int *c0); //操作指令
//**************************************
int main(void)
{
    redir(); //redirection
//**************************************
/* Work Space*/
    int i, j;
    int kase = 0;
    int q;
    int r0, c0;

    while(scanf("%d%d%d", &r, &c, &n) == 3){
        for(i=0; i<n; i++){
            scanf("%s", cmd[i].c);
            if(cmd[i].c[0] == 'E'){
                scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
            }else{
                scanf("%d", &cmd[i].a);
                for(j=0; j<cmd[i].a; j++){
                    scanf("%d", &cmd[i].x[j]);
                }
            }
        }

        if(kase > 0){
            printf("\n");
        }
        printf("Spreadsheet #%d\n", ++kase);

        scanf("%d", &q);
        while(q--){
            scanf("%d%d", &r0, &c0);
            printf("Cell data in (%d, %d) ", r0, c0);
            if(!simulate(&r0, &c0)){
                printf("GONE\n");
            }else{
                printf("moved to (%d, %d)\n", r0, c0);
            }
        }
    }
//**************************************
    freopen("CON", "r", stdin); //取消重新導向
    freopen("CON", "w", stdout);

    printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
```

```
    system("pause");
    return 0; //the end...
}

void redir(void)
{
    freopen(IN, "r", stdin);
    freopen(OUT, "w", stdout);
}
//**************************************
/* Work Space*/
//操作指令
int simulate(int *r0, int *c0)
{
    int i;
    int j;
    int x;
    int dr, dc;

    for(i=0; i<n; i++){
        if(cmd[i].c[0] == 'E'){ //"EX"
            if(cmd[i].r1 == *r0 && cmd[i].c1 == *c0){
                *r0 = cmd[i].r2;
                *c0 = cmd[i].c2;
            }else if(cmd[i].r2 == *r0 && cmd[i].c2 == *c0){
                *r0 = cmd[i].r1;
                *c0 = cmd[i].c1;
            }
        }else{
            dr = dc = 0;
            for(j=0; j<cmd[i].a; j++){
                x = cmd[i].x[j];
                if(cmd[i].c[0] == 'I'){
                    if(cmd[i].c[1] == 'R' && *r0 >= x) dr++; //"IR"
                    if(cmd[i].c[1] == 'C' && *c0 >= x) dc++; //"IC"
                }else{
                    if(cmd[i].c[1] == 'R' && *r0 == x) return 0; //"DR" : 所在列被刪除
                    if(cmd[i].c[1] == 'C' && *c0 == x) return 0; //"DC" : 所在欄被刪除
                    if(cmd[i].c[1] == 'R' && *r0 > x) dr--; //"DR"
                    if(cmd[i].c[1] == 'C' && *c0 > x) dc--; //"DC"
                }
            }
            *r0 += dr;
            *c0 += dc;
        }
    }
    return 1;
}

//Input(IN) Sample
/*
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
```

```
6 5
0 0
*/
```